Orange and Decisions-at-Hand: Bridging Predictive Data Mining and Decision Support

Blaž Zupan[1,2], Janez Demšar[1], Michael W. Kattan[3], Mak Ohori[3], Markus Graefen[4], Marko Bohanec[5], and J. Robert Beck[2]

[1] Faculty of Computer and Information Science, University of Ljubljana, Slovenia
[2] Baylor College of Medicine, Houston, TX
[3] Memorial Sloan Kettering Cancer Center, New York, NY
[4] Department of Urology, University of Hamburg, Germany
[5] J. Stefan Institute, Ljubljana, Slovenia

Abstract. Data mining is often used to develop predictive models from data, but rarely addresses how these models are to be employed. To use the constructed model, the user is usually required to run an often complex data mining suite in which the model has been constructed. A better mechanism for the communication of resulting models and less complex, easy to use tools for their employment are needed. We propose a technological solution to the problem, where a predictive model is encoded in XML and then used through a Web- or Palm handheld-based decision support shell. This schema supports developer-to-user and user-to-user communication. To facilitate the communication between the developers we advocate the use of data mining scripts.

1 Introduction

Development of predictive models is one of the key areas within data mining and relies on various approaches including statistics, machine learning and intelligent data analysis. In our view, predictive data mining has two different aims: (1) the uncovering of hidden relationships and patterns in the data and (2) the construction of usable prediction models. While authors that report on predictive data mining most often address the first issue, the second aim is usually covered solely in terms of assessment of predictive accuracy. Cases where models are indeed put into practice are rare, and so are the analysis of how the newly developed models improve some decision-making process [2].

Decision support systems most often rely on some incorporated predictive model. The above-described problem can thus be stated as bridging the gap between predictive data mining and decision support. The aspect of this bridge that we address in this paper is technological and deals with technical means that support the communication between model developers and users.

First, there is an issue of communication between the developers of the model. Their communication should incorporate the information on what procedures were executed on the data to derive the predictive model. While, at a tactical level, CRISP guidelines [4] provide a well-accepted framework to organize such

a description, their technical aspects are perhaps best covered by some script language that is supported by the data mining suite of choice. Data mining script can effectively and concisely specify the procedures that were executed, and when appropriately annotated through the use of comments, may have also a documental value. Developers should be allowed to easily modify the scripts, incorporate additional methods and approaches, and communicate the scripts to other members of data analysis team. Many modern data mining tools support scripting, giving them a particular advantage to the purely GUI-based systems.

Next, there is a communication between developers and users. Most often, the programs for data analysis often tend to be large complex for a non-specialist user. While targeting data analysis, we believe that these environments are inappropriate for decision support. Decision support may often require much less resources (computing power, methods and procedures) than data analysis, and, when using specific types of predictive models, can be implemented within small, easy to use programs. While the means of encoding predictive models (say, in XML) are emerging, so should the corresponding tools for decision support be developed that are lightweight, easy to use, and inexpensive.

Finally, there is a user-to-user communication. A satisfied user will often be willing to share her experience on using the model with her colleague, thereby promoting the utility of the model and increasing its practical value.

The technology should thus allow a seamless exchange of predictive models and provide tools for their effective and appropriate use. With emergent mobile devices, the employment of predictive models should not be limited to PCs. For instance, in medicine, decisions often take place where there is no PC immediately available.

In this paper, we show how a specific data mining suite called Orange can be used to support different phases of CRISP guidelines, resulting in a script that reads the data, develops a predictive model, and stores it in XML format. The core of this paper (and perhaps its major originality) is our proposal of technology called Decisions-at-Hand that allows the user to apply XML-encoded model through either a Web- or a Palm handheld-based decision support shell. We illustrate the proposed approach through the analysis of prostate cancer patient data, targeting the development of the model that can be used in clinical practice and that, based on preoperative findings, predicts the probability of cancer being organ confined. We highlight the importance of standards for representing predictive models, including the emerging standard PMML (Predictive Model Markup Language), and give some proposals for further work.

2 Predictive Data Mining in Orange: A Case Study on Prostate Cancer Data

A major goal in predictive data mining is to construct a predictive model. Most often, this is a four-step procedure that includes familiarization with the data, data preprocessing, modelling and evaluation of performance of constructed models.

Orange is a tool that can support all the phases of the CRISP process. In this section, we show its application on the problem from urology. Based on retrospective data from Memorial Sloan Kettering Cancer Center in New York, we build a predictive model that, given a set of pre-operative findings, tries to estimate the probability that the tumor is organ confined. The motivation for this task is in the clinically usefulness of such model, because organ confined disease is an important endpoint for surgical decision-making. It gives some indication of the curability of the prostate cancer, as well as guidelines for the surgical technique to be used (*e.g.*, nerve sparing vs. non nerve-sparing).

The data set that we have used is comprised of retrospective records about 1269 patients, of which for 768 (60.5%) their tumor was found to be organ confined. Prior to operation, ten commonly used predictors (Table 1) were recorded for each patient. Our task was therefore to examine if these features contain sufficient information to predict the probability of organ-confined tumor, and if so, to construct a corresponding prognostic model.

Table 1. Features describing prostate cancer patients

Feature	Description
PSA	preoperative PSA
clinstage	clinical stage (T1c, T2a, T2b, T2c, T3)
bxno	number of biopsy
posibxno	number of cancer positive cores
bx45posi	number of cores with Gleason grade 4 or 5
gg1	primary Gleason grade in RPX specimens
gg2	secondary Gleason grade in RPX specimens
dre	digital rectal examination; 0:neg, 1: pos
totalca	total length of cancer in all cores
totalbx	total length of all biopsy cores

2.1 Familiarization with the data

The easiest way to "get know" the data is to use visualization. Orange's suite includes Orange.First, an easy-to-use tool to explore the data. It targets visualization of classified data, and for instance supports different visualizations of correlations of features and outcome. For example, correlation between a feature and the outcome can be observed in the form of bar graphs giving the distribution of outcomes at different values of the feature. Fig. 1 shows that the probability of tumor being organ-confined is at the highest (around 0.9) at a moderate value of *posibxno*. Patients with higher values have lower probabilities of organ-confined tumor, though the number of such cases is small and the confidence intervals show that this trend probably cannot be proven by our data.

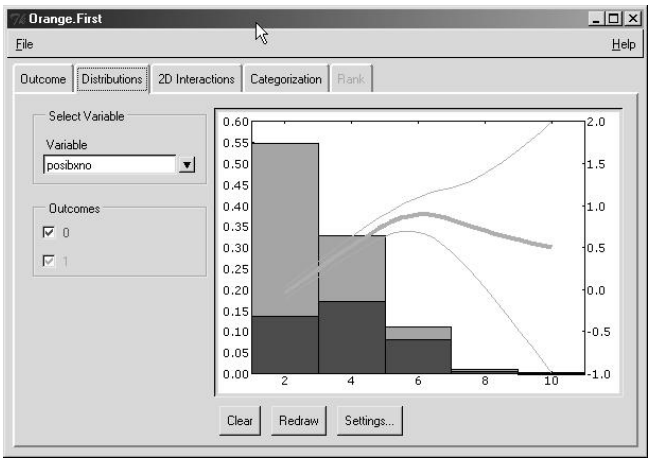

Fig. 1. Probabilities of organ-confined tumor for patients with different number of cancer positive cores

2.2 Preprocessing

Preprocessing can be done automatically or manually. In our case, we need to categorize the features and select those that will be included in the model. We usually let the computer program make a proposition, but then ask the domain expert to make the final decision.

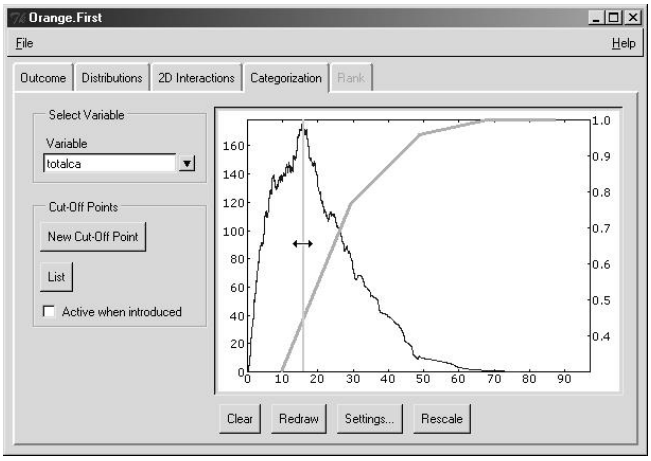

Fig. 2. Categorization with Orange.First

Orange.First can assist in a manual search for a suitable categorization. It draws a graph (Fig. 2) depicting a probability of certain class at different values of

the attribute and, at each point, a chi-square difference between the neighboring intervals if this point would be used as a cut-off point. After setting a cut-off point, the chi-square curve adjusts so that it shows the difference for additional cut-off points.

Orange supports different automatic algorithms for determining cut-off points, such as equidistant categorization, quartile categorization, entropy-MDL based categorization. The latter [5] is especially interesting since it does not require the user to set the number of categories in advance. As a side effect, if it finds no cut-off point the feature is considered useless and should be discarded.

While we are still working on a graphical user interface for this task, writing a fairly simple Orange script in Python programming language can easily perform these operations:

```
import orange
origData = orange.ExampleTable("confined.tab")
import orngDisc
data = orngDisc.discretize(origData)
```

For the features *bxno* and *totalbx* the categorization did not find suitable cut-off points and suggested that this features may not be useful for prediction model. They were therefore removed by `orngDisc.discretize`. The remaining continuous-valued features were categorized by using cut-off points shown in Table 2.

Table 2. Categorization of features

Feature	Cut-Off Points
PSA	6.04, 14.40, 31.45
bx45posi	0.00, 1.00
Posibxno	2.00
Totalca	3.50, 15.85, 35.20

In Orange, an expert may edit cut-off points manually. After this is done, we can assess the quality of the attributes by measures like information gain or ReliefF [8]:

```
relieff=orange.MeasureAttribute_relief(k=20, m=50)
orngDisc.measure(data, relieff)
```

The advantages of ReliefF are that it does not suffer for myopia (a tendency of measures from information theory to underestimate the attributes that are not immediately useful) and that it assigns negative grades to attributes that it finds of no use. We can thus additionally discard the attributes that are ranked negative. There are no such attributes for our data (Table 3), though *posibxno*, *gg2* and possibly also *clin-stage* and *dre* show a low ReliefF, so the expert should consider whether they should be used for modelling or not.

This concludes the preprocessing; we removed several attributes and used the automatic categorization for the others.

Table 3. Features ranked by ReliefF

Feature	ReliefF
bx45posi	0.041
PSA	0.036
totalca	0.022
gg1	0.016
dre	0.009
clinstage	0.009
gg2	0.002
posibxno	0.001

2.3 Modelling

We used three machine learning methods, which are usually successfully applied to medical data: naive Bayesian modelling, and two similar algorithms for induction of classification trees: C4.5 [10] and Orange's implementation of classification trees. The following code constructs the modelling objects:

```
import orngBayes
import orngTree

bayes = orngBayes.BayesLearner()
bayes.name = 'Naive Bayes'

c45 = orange.C45Learner()
c45.commandline("-s")
c45.name = 'C45'

tree= orngTree.TreeLearner(minExamples=5.0, maxMajor=0.8, minLeaf=5.0)
tree.name = 'Orange Tree'
```

In Orange, a model is constructed by calling one of the just constructed objects with modelling data (patient descriptions) as a parameter. For instance, to construct a naive Bayesian classifier from our complete (and preprocessed) data set, a command `bayesmodel=bayes(data)` will suffice. For the purpose of evaluation, however, a cross-validation schema is used instead of constructing a single model.

2.4 Evaluation

In order to evaluate the models, we used ten-fold cross validation technique: data is divided into ten folds and the modelling is repeated ten times. Each time, we use nine folds for modelling and the remaining fold for testing the prediction.

There are many different statistics with which we can measure the fitness of models. The most common in machine learning is classification accuracy, the proportion of testing examples for which the outcome has been correctly predicted. By our experience, medical experts prefer the "area under ROC"

statistics, which is in interpretation a discrimination measure and, drawing two cases with different outcomes (organ-confined cancer, cancer that was not organ-confined), estimates the proportion of such cases where a model would correctly assign a higher probability of organ confined cancer to the case with cancer that was indeed organ-confined. The following code computes and prints both:

```
import orngEval
learners = (bayes, c45, tree)
results = orngEval.CrossValidation(learners, data)
cdt = orngEval.computeCDT(results)

print "Learner         CA     aROC"
for i in range(len(learners)):
    print "%-15s" % learners[i].name,
    print "%5.3lf" % orngEval.CA(results)[i],
          " %5.3lf" % orngEval.aROC(cdt[i])[7]
```

These two statistics are shown in Table 4. The naive Bayesian model has the highest accuracy and aROC. Although this is a result of a rather preliminary study, the aROC achieved by naive Bayes is, within participating institutions, the highest obtained for this problem and (by expert's opinion) likely better than any alternative, including clinical judgment.

Table 4. Accuracy and area under ROC of the models

Learner	CA	aROC
Naive Bayes	0.755	0.816
C45	0.735	0.776
Orange Tree	0.741	0.766

3 Encoding of Decision Models in XML

Being satisfied with the results of the evaluation, we can now choose our modelling technique, construct a predictive model from the complete data set, and output it to an XML file:

```
bayesmodel=bayes(data)
bayesmodel.saveAsXML("confined.xml")
```

The final model is defined with a little bit of editing of our XML file `confined.xml`, mainly to change the variable names to make them more descriptive, and to add some descriptive text. The file starts with some general information about the model and authors:

```
<?xml version="1.0" ?>
```

```
<model name="Organ Confined Prostate Cancer">
<description>
 Based on preoperative predictors computes the probability that prostate
 cancer is organ-confined.
</description>
<author>M. W. Kattan, B. Zupan, J. Demsar</author>
<date>May 2001</date>
<outcome>Probability of Prostate Cancer to be Organ-Confined</outcome>
```

The description of the model continues with the definition of predictive variables, describing their type and other information relevant for the data entry form (for our illustration, just the first two variables are listed):

```
<variables>
  <var>
    <name>PSA</name>
    <type>categorical</type>
    <input>pulldown</input>
    <values>&lt;=6.04;(6.04, 14.40];(14.40, 31.45];&gt;31.45</values>
  </var>
  <var>
    <name>Clinical Stage</name>
    <type>categorical</type>
    <input>pulldown</input>
    <values>T1c;T2a;T2b;T2c;T3a</values>
  </var>
  ...
</variables>
```

Lastly, the model itself is defined by means of stating a priory class probabilities and conditional probabilities that are required for the computation of probability of the outcome using naive Bayesian formula:

```
<modeldefinition type="naivebayes">
  <classprobabilities>0.605; 0.395</classprobabilities>
  <contingencymatrix>
    <conditionalprobability attribute="PSA">
      0.244,0.756; 0.384,0.616; 0.640,0.360; 0.898,0.102
    </conditionalprobability>
    <conditionalprobability attribute="Clinical Stage">
      0.265,0.735; 0.373,0.627; 0.642,0.358; 0.667,0.333; 0.838,0.162
    </conditionalprobability>
    ...
  </contingencymatrix>
</modeldefinition>
```

4 Decisions-at-Hand: A Decision Support Shell Approach

Decisions-at-Hand schema currently supports naive Bayesian and logistic regression models. It is implemented through a Web-enabled server-based decision support application and software that runs on a Palm handheld computer. While,

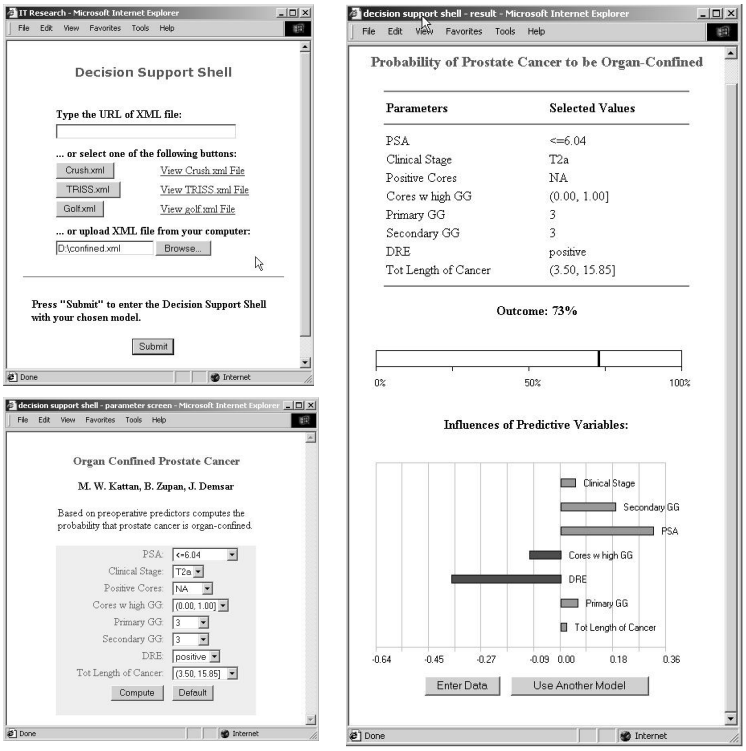

Fig. 3. Web interface of Decisions-at-Hand: entry to Web-based decision support shell (top left), specification of predictive factors (bottom left), and outcomes (right).

based on the current experience from Memorial Sloan Kettering Cancer Center [6], we expect that in clinical practice Palm solutions will be better accepted, the Web-based interface may especially be useful in the decision support system prototyping and testing phase.

Three snapshots of the Web interface are given in Fig. 3. The user starts by uploading a decision model (`confined.xml`), and can then enter the patient-specific data. Finally, the page with outcomes shows the probability of organ-confined tumor and provides additional information on an influence of specific predictor. The influence can be either positive or negative, depending on the contribution of a specific factor towards the increase or decrease of probability of outcome in respect to a priory probability. For this, we have implemented the naive Bayesian approach as suggested in [7] and previously applied for medical diagnosis in [11].

The user interface on Palm is conceptually similar to the one available through the Web. XML models are transferred to Palm through synchronization with PC. While the user can choose any downloaded model, a recently used one on Palm is displayed immediately after the invocation of the program. At present, only the

shell for logistic regression models is implemented (LogReg program), and we are currently working on the support for other types of predictive models. Few snapshots of Palm interface are given in Fig. 4, this time showing an example of utility of the logistic regression-based decision support model to determine the severity of crush injury [1]. Decisions-at-Hand schema on Palm is described in further detail in [13].

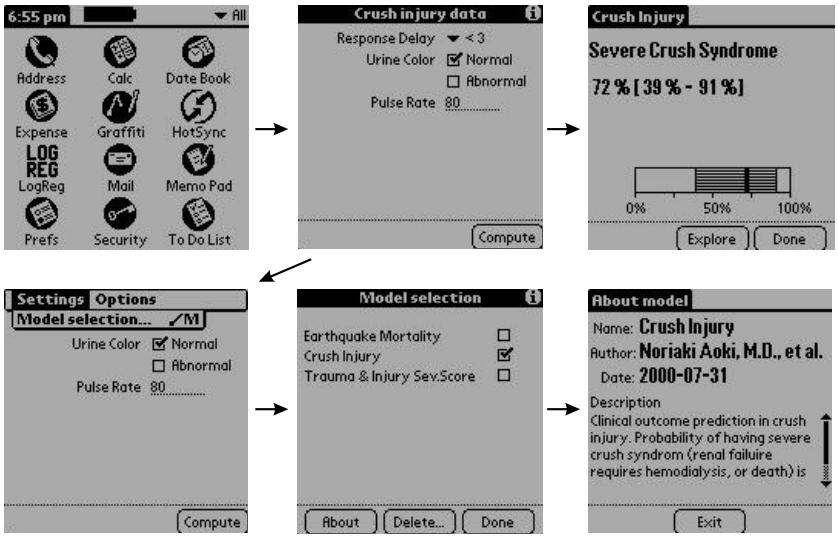

Fig. 4. Snapshots from decision support shell on Palm.

5 Towards the Standardization of Model Interchange

A distinguishing strength of the Decisions-at-Hand schema lies in its ability to use predictive models developed by some external data analysis or data mining system, such as Orange. This requires a standardized model representation that is easily exchanged and manipulated by both humans and machines. As illustrated in previous section, our representation is based on XML using a proprietary document type definition that facilitates the representation of naive Bayesian and logistic regression models.

Recently, an important initiative has been raised by the Data Mining Group (DMG, `http://www.dmg.org/`) to provide a vendor-independent open standard of defining data mining models. DMG is "an independent, vendor-led group which develops data mining standards", which currently includes seven members: Angoss, IBM, Magnify, NCR, Oracle, SPSS, and the National Centre of Data Mining at the University of Illinois, Chicago. Their emerging standard is called Predictive Model Markup Language (PMML, `http://www.dmg.org/html/pmml_v1_1.html`). It is based on XML, and facilitates the exchange of models

between different vendors' applications in order to visualize, analyze, evaluate or otherwise use the models. The most recent version of PMML is 1.1, which supports the following types of models: polynomial and general regression, decision trees, center and density based clusters, association rules, and neural nets.

Unfortunately, none of the models used in Decisions-at-Hand are currently supported by PMML. This is a big obstacle, and we urge for incorporating both naive Bayesian and logistic regression models into PMML. For applications in medicine, some other model types would be useful as well: decision tables (interestingly, perhaps, the most known and used decision models in the area of prostate cancer are decision tables [9]), hierarchical multi-attribute models [3], and survival prediction models [12].

After a basic support for some model type has been provided in PMML, it seems relatively straightforward to translate a proprietary XML format into PMML, or even completely replace the former with the latter. The formats are similar and contain similar elements: general information about the model (referred to as *Header* in PMML), definition of variables (*Data Dictionary* and *Mining Schema*), and the model itself. Some elements related to user interface, such as `<input>pulldown</input>`, will need to employ PMML's *Extension Mechanism*.

Notice also that there are other, commercial attempts, like IBM's Intelligent Miner Scoring schema (see `www-4.ibm.com/software/data/iminer/scoring/`), to use PMML as a communication mechanism between model generation and model utilization. Besides being free, perhaps the main difference between these and approach proposed in this paper is (1) in simplicity and small size of decision support shells and (2) in the exclusive platforms – Web and handheld computers – that we are targeting.

6 Conclusion

Data mining is often concerned with development of predictive models. In order for these to be really used in daily practice, they have to be seamlessly integrated within easy-to-use decision support systems. The authors of this paper believe that appropriate technology is one of the key factors that may help to advance the acceptance and use of predictive models in practice, and facilitate the communication be-tween the developers and users. In particular, we believe that crafting the appropriate easy-to-use and readily (freely?) available decision support shells may help in this endeavor. For this, we propose a schema where a predictive model is developed separately within some data mining or data analysis suite (in our case, in Orange), while for decision support a decision shell is to be used either through a Web-based interface or on a handheld computer.

The utility of the Orange data mining suite and applications within our Decisions-at-Hand schema for development of prognostic models and decision support were presented in the paper. They are both freely available at the Web sites `http://magix.fri.uni-lj.si/orange` and `http://magix.fri.uni-lj.si/palm`.

Acknowledgement

This work was supported, in part, by the grant RPG-00-202-01-CCE from the American Cancer Society (MWK, BZ, JD), by the Program Grant from Slovene Ministry of Science and Technology (JD, BZ, MB), and by the EU project SolEuNet, IST-11495 (MB). We would like to thank Aleš Porenta, Xiaohuai Yang and Gaj Vidmar for their help in the implementation of Palm and Web-based applications, and Noriaki Aoki, M. D., for his initiative, expertise and help in development of predictive model for crush injury syndrome.

References

[1] N. Aoki, J. R. Beck, and E. A. P. *et al.* The risk factors of crush injury patients for renal failure, hemodialysis and death. *Prehospital Disaster Med*, 14:S54–55, 1999.

[2] R. Bellazzi and B. Zupan. Intelligent data analysis in medicine and pharmacology: A position statement. In *IDAMAP-98*, pages 1–4, Brighton, UK, 1988.

[3] M. Bohanec, B. Zupan, and V. Rajkovič. Applications of qualitative multi-attribute decision models in health care. *International Journal of Medical Informatics*, 58–59:191–205, 2000.

[4] P. Chapman, J. Clinton, T. Khabaza, T. Reinartz, and R. Wirth. The CRISP-DM process model (available at http://www.crisp-dm.org). Technical report, CRISP-DM consortium, 1999.

[5] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029, Chambery, France, 1993. Morgan-Kaufmann.

[6] M. W. Kattan, T. M. Wheeler, and P. T. Scardino. Postoperative nomogram for disease recurrence after radical prostatectomy for prostate cancer. *Journal of Clinical Oncology*, 17(5):1499–1507, 1999.

[7] I. Kononenko. Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence*, 7:317–337, 1993.

[8] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In F. Bergadano and L. de Raedt, editors, *Proceedings of the European Conference on Machine Learning*, pages 171–182. Springer-Verlag, 1994.

[9] A. W. Partin, J. Yoo, D. Chan, J. I. Epstein, and P. C. Walsh. The use of prostate specific antigen, clinical stage and gleason score to predict pathological stage in men with localized prostate cancer. *Journal of Urology*, 50:110–114, 1993.

[10] R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[11] I. Zelič, I. Kononenko, N. Lavrač, and V. Vuga. Induction of decision trees and bayesian classification applied to diagnosis of sport injuries. *J. Med. Syst.*, 21:429–444, 1997.

[12] B. Zupan, J. Demšar, M. W. Kattan, J. R. Beck, and I. Bratko. Machine learning for survival analysis: A case study on recurrence of prostate cancer. *Artificial Intelligence in Medicine*, 20:59–75, 2000.

[13] B. Zupan, A. Porenta, G. Vidmar, N. Aoki, I. Bratko, and J. R. Beck. Decisions at hand: A decision support system on handhelds. In *Proc. Medinfo-2001 (in print)*, 2001.